

(19)



Europäisches Patentamt
European Patent Office
Office européen des brevets



(11)

EP 1 191 760 B1

(12)

EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention
of the grant of the patent:
10.08.2005 Bulletin 2005/32

(51) Int Cl.7: **H04L 29/06**, H04L 29/14

(21) Application number: **01120765.1**

(22) Date of filing: **07.09.2001**

(54) **Apparatus and method for header decompression**

Vorrichtung und Verfahren für Kopfdekomprimierung

Appareil et procédé permettant la décompression d'en-têtes

(84) Designated Contracting States:
DE FI FR GB SE

(30) Priority: **11.09.2000 JP 2000274615**
01.12.2000 JP 2000367514

(43) Date of publication of application:
27.03.2002 Bulletin 2002/13

(60) Divisional application:
04009502.8 / 1 447 956

(73) Proprietor: **MATSUSHITA ELECTRIC INDUSTRIAL
CO., LTD.**
Kadoma-shi, Osaka-fu (JP)

(72) Inventors:
• **Hata, Koichi**
Katano-shi, Osaka-fu (JP)
• **Miyazaki, Akihiro**
Sakai-shi, Osaka-fu (JP)
• **Imura, Koji**
Machida-shi, Tokyo-to (JP)

• **Ido, Daiji**
Yokohama-shi, Kanagawa-ken (JP)

(74) Representative: **Lang, Johannes, Dipl.-Ing.**
Bardehle Pagenberg Dost
Altenburg Geissler Isenbruck
Postfach 86 06 20
81633 München (DE)

(56) References cited:
• **CARSTEN BORMANN ET. AL.: "Robust Header
Compression (ROHC)" INTERNET DRAFT,**
[Online] 14 July 2000 (2000-07-14), pages 1-34,
XP002247470 Retrieved from the Internet:
<URL:http://www.watersprings.org/pub/id/dr
aft-ietf-rohc-rtp-01.txt> [retrieved on
2003-07-14]
• **ROHC MAILING LIST ARCHIVE, [Online] 10 July**
2000 (2000-07-10), pages 1-9, **XP002247471**
ROHC mailing list archive Retrieved from the
Internet:
<URL:http://www.cdt.luth.se/rohc/msg00559.
html> [retrieved on 2003-07-14]

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

EP 1 191 760 B1

Description

BACKGROUND OF THE INVENTION

Field of the Invention

[0001] The present invention relates to methods for header compression/decompression in packet transmission and, more specifically, to a method for header compression/decompression, where reference information is requested to be updated when an error occurs in packet transmission.

Description of the Background Art

[0002] Typical protocols recently known for data transmission over the Internet include TCP/IP (Transmission Control Protocol/Internet Protocol) and UDP/IP (User Datagram Protocol/Internet Protocol). In data transmission under such transmission protocol over a low and midrange bit rate transmission path, the header specified by TCP, UDP, IP, or other protocols is larger in size, disadvantageously causing overhead associated with communication. For example, to transmit 10-byte data under UDP/IP, the transmitting side has to add a 28-byte header to the original data, resultantly forming a 38-byte packet, which is approximately four times larger in size than the original data. If such increase in size happens quite often, the transmission path is substantially decreased in effective speed.

[0003] In order to reduce communication overhead caused by the header, a header compression scheme developed by V. Jacobson and defined in RFC 1144 and RFC2508 has been known. In this scheme, among the fields of the header included in the packet, transmitted are only any field changed in value from the one included in the previous packet. Such field changed in value are not so many in the header, and therefore, in this scheme, header compression is successfully carried out. This header compression scheme, however, is a standard for wired communication with a low transmission error rate, as shown in FIG. 5, and is not efficient for a transmission path with a high transmission error rate.

[0004] FIG. 6 shows a communication network for wireless terminals over a cellular phone network such as W-CDMA. In recent years, the number of users of such communication network is rapidly growing. The communication network of FIG. 6 includes a wireless transmission section where errors frequently occur. To reduce overhead caused by the header in a wireless section, one header compression scheme is known as ROHC (RObust Header Compression) studied by IETF (Internet Engineering Task Force). The detail of ROHC is described in "draft-ietf-rohc-rtp-00.txt (29 June 2000)", and in the successive "draft-ietf-rohc-rtp-01.txt (14 July 2000)". Further details to said IETF drafts can be found in a posting to the IETF ROHC WG mailing list,

(10 July 2000), "Re: AW: [rohc] SO packets with keywords", available under URL: <http://www.cdt.luth.se/rohc/msg0059.html>, which discloses that a refresh procedure is triggered if two consecutive CRCs indicate an error.

[0005] In ROHC, for data compression at the transmitting side (compressing side) and data decompression at the receiving side (decompressing side), reference information is shared by both sides for reference. That is, the reference information referred to for data compression at the transmitting side is also referred to for data decompression at the receiving side. By sharing the reference information, data decompression can be correctly achieved. FIG. 7 shows one example of data transmission adopting ROHC.

[0006] In FIG. 7, at the start of data transmission, the transmitting side and the receiving side each have held correct reference information α . Consider first a case where the transmitting side transmits a header $H1$ and data $D1$ to the receiving side. Before transmission, the transmitting side carries out data compression on the header $H1$ by referring to the reference information α . Here, the header $H1$ and a compressed header $H'1$ for transmission to the receiving side have such a relation as represented by the following equation (1).

$$H'1 = H1 * \alpha \quad (1).$$

[0007] In the above equation (1), an operation represented by $*$ varies for each field of the header to be compressed. For example, the operation is so carried out as follows: the field does not vary if representing a UDP port number; the field is generally increased in value by 1 if representing an RTP sequence number; and the field is increased in value by 50 if representing an RTP timestamp.

[0008] As such, the reference information α includes all information required for compression of each field as described above. Therefore, if the receiving side holds the correct reference information α having the same contents as that held in the transmitting side, the receiving side can correctly decompress the received compressed header $H'1$ into the original header $H1$, thereby obtaining the correct header $H1$ and data $D1$. Similarly, headers and data $H2$ and $D2$, $H3$ and $D3$, and $H4$ and $D4$ are transmitted after each header is compressed by referring to the reference information α .

[0009] Next, consider a case where the reference information is changed. FIG. 8 shows an example of data transmission where the reference information is changed during the transmission. In FIG. 8, after the header $H2$ and the data $D2$ are transmitted, the reference information is changed from α to β , and the header $H3$ is compressed by referring to the changed reference information β .

[0010] For example, assume that the RTP timestamp of the header to be transmitted is increased by 50, but,

at the time of transmission of the data D3, such increase is changed to by 100. Under this assumption, the transmitting side changes the reference information α held so far containing that "The RTP timestamp is increased by 50" into the reference information β containing that "The RTP timestamp is increased by 100". To update the reference information, as shown in FIG. 8, the receiving side refers to update information further provided to the compressed header to be transmitted (here, a header H'3).

[0011] In some cases, the reference information may be updated even if the update information is not explicitly transmitted. One example header compression scheme taken in such cases is briefly described below. In the compressed header, the sequence number is assigned 4 bits capable of representing integers from 0 to 15, but not 16 or more. Therefore, any integer N equal to 16 or more is represented by $N \bmod 16$. Thus, the receiving side finds the sequence number by using an equation $L * 16 + (\text{received sequence number})$, where L is incremented by 1 whenever the received sequence number is changed from the maximum value (here, 15) to the minimum value (here, 0). Here, the update information is not explicitly transmitted. Instead, when the sequence number becomes larger than the maximum value, the reference information is updated on both sides.

[0012] FIG. 9 is a block diagram showing the structure of a header decompression apparatus that achieves the header decompression as described above.

[0013] In FIG. 9, a header decompression apparatus 1007 includes a packet output unit 1001, an error detector 1002, a header decompressor 1003, a packet receiver 1004, a reference information manager 1005, and an update request unit 1006.

[0014] The packet receiver 1004 receives a header-compressed packet from a transmitting side, and outputs the packet to the header decompressor 1003. The header decompressor 1003 refers to reference information managed by the reference information manager 1005 to decompress the compressed header, and outputs the header-decompressed packet to the error detector 1002. If the compressed header is provided with update information, the header decompressor 1003 updates the reference information managed by the reference information manager 1005 with the update information provided to the compressed header. The error detector 1002 detects any error in the header-decompressed packet. If not detecting an error, the error detector 1002 outputs the correctly-decompressed packet to the packet output unit 1001. If detecting an error, the error detector 1002 discards the packet as not having been correctly decompressed. The update request unit 1006 receives a notification that an error is detected by the error detector 1002, and transmits an update request to the transmitting side. Specifically, according to the above document, "draft-ietf-rohc-rtp-00.txt (29 June 2000)", the update request unit 1006 transmits a NACK

packet. The reference information manager 1005 manages the reference information for header decompression. The packet output unit 1001 outputs the header-decompressed packet.

[0015] As such, the header decompression apparatus 1007 detects any error in the compressed header. Here, typically, the compressed header is provided with a CRC (Cyclic Redundancy Code) for determining whether the header-decompressed packet has any error or not. Therefore, any error that occurred in the compressed header or a payload due to noise during wireless transmission can be detected, and the erroneous packet can be discarded.

[0016] FIG. 10 shows one example of data transmission where an error occurs due to noise during wireless transmission. In FIG. 10, a header H2 is compressed to be a header H'2, and the header H'2 and data D2 are wirelessly transmitted. During the wireless transmission, noise or other factors affect the compressed header H'2, causing an error, which is denoted by a dotted cross in FIG. 10. As a result, as denoted by a solid cross in FIG. 10, the header cannot be correctly decompressed at the receiving side, and therefore the entire packet is discarded.

[0017] Such error as described above may occur also in the compressed header with the update information provided thereto. FIG. 11 exemplarily shows a case where an error occurs in the header with the update information provided thereto, and the reference information is erroneously updated. In FIG. 11, a header H3 is compressed to be a header H'3, and the header H'3 and data D3 is wirelessly transmitted. During the wireless transmission, noise or other factors affect the update information provided to the compressed header H'3, causing a change in the update information, which is denoted by a dotted cross in FIG. 11. Therefore, at the receiving side, the reference information is erroneously updated to reference information β' , based on the changed update information, and the header H3 decompressed by referring to the erroneous reference information is not the same as the original header H3 before compression at the transmitting side. This also applies to the following headers H4 and thereafter. As a result, as denoted by a solid cross in FIG. 11, the header cannot correctly be decompressed at the receiving side, and is generally regarded as having an error. Therefore, the entire packet is discarded.

[0018] In some cases, however, the header is not regarded as having an error even if it has not been correctly decompressed, and therefore the packet is not discarded. FIG. 12 exemplarily shows a case where a packet is not regarded as having an error even if the reference information is erroneously updated. In FIG. 12, the reference information is erroneously updated at the receiving side to become receiving-side reference information α' , which is different from the reference information α at the transmitting side. As a result, headers H1 to H4 are erroneously decompressed at the receiving

ing side. Therefore, in general, CRC errors occur and the entire packet are discarded. However, according to principles of CRC, not all errors cannot be detected, and any erroneously-decompressed header may be accidentally determined as being correct. In FIG. 12, a packet containing the header *H3* and the data *D3* is accidentally determined as being correct, and is not discarded.

[0019] As stated above, even if no error is detected, there may be a decompression error caused by noise or erroneous reference information. Also, even if one decompression error is detected, it is impossible to tell the cause of the decompression error, that is, either noise or erroneous reference information. Therefore, according to the above background art, an update request typified by NACK is transmitted whenever an error occurs. Such update request, however, is unnecessary when the reference information is correct and the error is caused only by noise. Thus, according to the background art, the more errors caused by noise, the more unnecessary update requests are transmitted, and therefore the lower header compression efficiency becomes.

SUMMARY OF THE INVENTION

[0020] Therefore, an object of the present invention is to provide a header decompression method in which only a necessary request for updating reference information is made based on the state of an error in a header-decompressed packet.

[0021] This is achieved by a header decompression apparatus as defined in independent claim 1, a header decompression method as defined in independent claim 3, a computer-readable recording medium with a program recorded therein, the program being executed in a computer system for carrying out header decompression as defined in independent claim 5, and a computer program product for carrying out header decompression as defined in independent claim 6. Further preferred embodiments are the subject of dependent claims 2 and 4.

[0022] The present invention has the following features to attain the object above.

[0023] A first aspect of the present invention is directed to a header decompression apparatus for decompressing a compressed header of a packet for transmission by referring to reference information being the same as reference information referred to for header compression by a transmitting side. In the apparatus, a packet receiver receives the packet from the transmitting side. A reference information manager stores and manages the reference information. A header decompressor is provided with the received packet, and carries out header decompression by referring to the reference information stored in the reference information manager.

[0024] An error detector detects an error in the packet including the decompressed header. A counter/storage counts and stores the number of errors detected by the

error detector. An update request unit transmits, to the transmitting side, update information for updating the reference information, when determining, that the reference information stored in the reference information manager should be updated. When the update information is transmitted from the transmitting side, the reference information manager updates the reference information stored in the reference information manager based on the transmitted update information.

[0025] The counter/storage includes a successive error counter and a successive decompression success counter. The successive error counter counts and stores the number of times *X* the packet having the error detected by the error detector successively appeared. The successive decompression success counter counts and stores the number of times *Y* the packet having no error successively appeared after the error detector detects the error.

[0026] Moreover, the update request unit determines, based on values of *X* and *Y* counted by the counter/storage, whether the reference information stored in the reference information manager should be updated.

[0027] As described above, in the first aspect, based on the number of decompression errors and decompression successes that successively appeared in the receiving side, an update request is transmitted to the transmitting side. Thus, the number of packets to be discarded due to error in the reference information can be reduced, and efficient packet transmission can be achieved.

[0028] According to a second aspect, in the first aspect,

when $X \geq Y$, the update request unit determines that the reference information stored in the reference information manager should be updated.

[0029] As described above, in the second aspect, when $X \geq Y$, a large number of successive decompression errors have occurred, and therefore there is a high possibility that the reference information has an error. Thus, the number of packets to be discarded due to error in the reference information can be reduced, and efficient packet transmission can be achieved. These and other objects, features, aspects and advantages of the present invention will become more apparent from the following detailed description of the present invention when taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0030]

FIG. 1 is a block diagram showing the structure of a data compression apparatus 607 according to a first embodiment of the present invention;
FIG. 2 is a block diagram showing the structure of a data decompression apparatus 709 according to the first embodiment of the present invention;

FIG. 3 is a diagram exemplarily showing the state of decompression and values of X and Y when a packet $P1$ to a packet $P11$ are received;

FIG. 4 is a block diagram showing the structure of a data decompression apparatus 908 according to a second embodiment of the present invention;

FIG. 5 is a diagram showing a header compression section in wired communication;

FIG. 6 is a diagram showing a header compression section in wireless communication;

FIG. 7 is a diagram exemplarily showing data transmission under a conventional header compression scheme of ROHC;

FIG. 8 is a diagram showing an example of data transmission where reference information is changed during the transmission in conventional header compression;

FIG. 9 is a block diagram showing the structure of a header decompression apparatus that achieves the header decompression in conventional art;

FIG. 10 is a diagram showing one example of data transmission where an error occurs due to noise during wireless transmission in the conventional header compression;

FIG. 11 is a diagram exemplarily showing a case where, in the conventional header compression, an error occurs in the header with reference update information provided thereto, and the reference information is erroneously updated; and

FIG. 12 is a diagram exemplarily showing a case where, in the conventional header compression, a packet is not regarded as having an error even if the reference information is erroneously updated.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

(First Embodiment)

[0031] A method for header compression/decompression according to a first embodiment of the present invention is realized by a header compression apparatus and a header decompression apparatus as shown in FIGS. 1 and 2, respectively. With reference to the drawings, the structure of each apparatus is described below.

[0032] FIG. 1 is a block diagram showing the structure of a header compression apparatus 607. In FIG. 1, the header compression apparatus 607 includes a packet input unit 601, a CRC provider 602, a header compressor 603, a packet transmitter 604, a reference information manager 605, and an update request receiver 606.

[0033] The packet input unit 601 outputs a received packet to the CRC provider 602. The CRC provider 602 provides the received packet with a CRC for output to the header compressor 603. The header compressor 603 refers to reference information managed by the reference information manager 605 to compress the head-

er of the received packet, and outputs the resultant packet to the packet transmitter 604. How to compress the header will be described later. The packet transmitter 604 transmits the received header-compressed packet to the receiving side. The reference information manager 605 manages the reference information referred to by the header compressor 603. The update request receiver 606 receives an update request from the transmitting side, and notifies the reference information manager 605 of the update request. The notified reference information manager 605 outputs the managed reference information to the header compressor 603, and instructs it to provide update information corresponding to the reference information to the compressed header.

[0034] FIG. 2 is a block diagram showing the structure of a header decompression apparatus 709. In FIG. 2, the header decompression apparatus 709 includes a packet output unit 701, an error detector 702, a header decompressor 703, a packet receiver 704, a successive decompression error counter (hereinafter, successive error counter) 705, a successive decompression success counter 706, a reference information manager 707, and an update request unit 708.

[0035] The packet receiver 704 receives, from the transmitting side, the header-compressed packet, and outputs it to the header decompressor 703. The header decompressor 703 refers to the reference information managed by the reference information manager 707 to decompress the compressed header, and outputs the resultant packet to the error detector 702. The error detector 702 detects a CRC error in the header-decompressed packet, and outputs, to the packet output unit 701, only any packet whose header has been correctly decompressed. The successive error counter 705 counts the number of successive decompression errors detected by the error detector 702. The successive decompression success counter 706 counts the number of successive decompression successes detected by the error detector 702. The update request unit 708 refers to the numbers counted by the successive error counter 705 and the successive decompression success counter 706 to determine whether an update request is required or not in a manner described later, and transmits the update request as required to the transmitting side. The reference information manager 707 manages the reference information for header decompression.

[0036] Note that, in the present invention, "successive" decompression errors mean that successive two or more error states are observed in a header, or that only one error state is observed therein, although the latter is not generally applicable to the meaning of the word "successive". Similarly, "successive" decompression successes means that successive two or more error-free states are observed in a header, or that only one error-free state is observed therein. How to determine "successive" states in the present invention will be de-

scribed later.

[0037] Described specifically below are the operation of each of the above structured header compression apparatus and header decompression apparatus according to the present embodiment. Note that a packet supplied to the apparatuses contains data, such as video and audio, with an RTP/UDP/IP header.

[0038] First, the operation of the header compression apparatus shown in FIG. 1 is specifically described. The packet input part 601 outputs an externally-inputted RTP/UDP/IP packet to the CRC provider 602. The CRC provider 602 computes a CRC for the entire packet, and provides the CRC to the packet.

[0039] The header compressor 603 refers to the reference information managed by the reference information manager 605 for header compression. In this header compression, if the header can be decompressed with its sequence number, the reference information is not updated, and only the sequence number is included in the header. If the header cannot be decompressed with its sequence number, update information of the reference information and the sequence number are included in the header. The packet transmitter 604 transmits the header-compressed packet to the receiving side.

[0040] The reference information manager 605 stores and manages the reference information referred to by the header compressor 603. Notified by the header compressor 603 of update, the reference information manager 605 updates the stored reference information. Notified by the update request receiver 606 that an update request has been received, the reference information manager 605 instructs the header compressor 603 to provide the update information to the header.

[0041] The update request receiver 606 receives an update request from the receiving side. On receiving the update request, the update request receiver 606 notifies the reference information manager 605 that the update request has been received.

[0042] Next, the operation of the header decompression apparatus 709 shown in FIG. 2 is specifically described. The packet receiver 704 receives the header-compressed packet transmitted from the packet transmitter 604 shown in FIG. 1, and outputs the packet to the header decompressor 703.

[0043] The header decompressor 703 refers to the reference information stored in the reference information manager 707 to decompress the compressed-header of the packet. The header-decompressed packet is outputted to the error detector 702. If the header is provided with update information, the header decompressor 703 notifies the reference information manager 707 of the update information.

[0044] The error detector 702 checks whether any error occurs or not in the header-decompressed packet by using the CRC. Also, the error detector 702 notifies the successive error counter 705 and the successive decompression success counter 706 of the presence or

absence of an error. If detecting any error, the error detector 702 discards the packet. If not detecting, the error detector 702 outputs, to the packet output unit 701, the packet with the CRC removed therefrom. The packet output unit 701 externally outputs the error-undetected RTP/UDP/IP packet.

[0045] The reference information manager 707 stores and manages the reference information required for header decompression. If the compressed header includes the update information, the reference information manager 707 updates the stored reference information with the update information coming from the header decompressor 703.

[0046] The successive error counter 705 counts the number of successive errors X based on the error detection results in the error detector 702. For example, if an error is detected in a packet, X is 1. Then, if an error is detected also in the next packet, X becomes 2. Then, if an error is detected still in the following packet, X is further incremented by 1 to become 3. If no error is detected, successiveness is interrupted, and the error counter 705 stops counting.

[0047] The successive decompression success counter 706 counts the number of decompression successes Y based on the error detection results in the error detector 702. For example, if no error is detected in a packet, Y is 1. Then, if no error is detected also in the next packet, Y becomes 2. Then, if no error is detected still in the following packet, Y is further incremented by 1 to become 3. If any error is detected, successiveness is interrupted, and the successive decompression success counter 706 stops counting.

[0048] The update request unit 708 determines whether to request update based on a set of X outputted from the successive error counter 705 and Y outputted from the successive decompression success counter 706. This determination operation is further described in detail with reference to FIG. 3.

[0049] FIG. 3 is a diagram exemplarily showing decompression operation and values of X and Y when packets $P1$ to $P11$ are received. In FIG. 3, assume that an error occurred in a packet $P12$ not shown.

[0050] In FIG. 3, an error is detected in the packet $P1$, and therefore $X = 1$. In the packet $P2$, no error is detected, and therefore X is held as 1 and $Y = 1$. Then, no error is detected from the packets $P3$ and $P4$, and therefore $X = 1$ and $Y = 3$. Then, when an error is detected in the packet $P5$, the values of X and Y for the packets $P1$ to $P4$ are regarded as a set representing the presence or absence of an error, and determined as $X = 1$, $Y = 3$. The values of X and Y for the packet $P5$ and thereafter are regarded as a new set. An error is detected in the packets $P5$ to $P8$, and therefore $X = 4$. No error is detected in the packets $P9$ to $P11$, and therefore $Y = 3$. Thereafter, when an error is detected in the packet $P12$, the values of X and Y for the packets $P5$ to $P11$ are regarded as a set, and determined as $X = 4$ and $Y = 3$.

[0051] As such, the update request unit 708 regards

the values of X and Y as a set representing successive decompression errors and subsequent decompression successes, and compares these values with predetermined values. By way of example, when X is larger than 10 and smaller than 2, it is determined that an error occurs in the reference information, and requests the transmitting side of updating the reference information. These predetermined values are example only, and not restrictive. If $X \geq Y$, a large number of successive decompression errors have occurred, and therefore there is a high possibility that the reference information has an error.

[0052] As such, the header decompression apparatus 709 according to the present embodiment requests the transmitting side of updating the reference information based on the number of successive decompression errors and successes at the receiving side. Thus, the number of packets discarded due to error in the reference information can be reduced, and efficient packet transmission can be achieved. Such capability of the header decompression apparatus 709 can be achieved in a general-purpose computer system. In this case, this capability is realized by a program executed in the computer system. The program is typically stored in a recording medium such as CD-ROM, or transmitted through a communication medium.

(Second Embodiment)

[0053] A method for header compression/decompression according to a second embodiment of the present invention is realized by a header compression apparatus and a header decompression apparatus as shown in FIGS. 1 and 4, respectively. Therefore, the header compression apparatus according to the second embodiment is the same in structure as the header compression apparatus 607 shown in FIG. 1, and not described herein. Described below is the structure of the header decompression apparatus according to the present embodiment with reference to FIG. 4.

[0054] FIG. 4 is a block diagram showing the structure of a header decompression apparatus 908. In FIG. 4, the header decompression apparatus 908 includes a packet output unit 901, an error detector 902, a header decompressor 903, a packet receiver 904, a decompression error detection result storage 905, a reference information manager 906, and an update request unit 907.

[0055] The packet receiver 904 outputs a header-compressed packet supplied from the transmitting side to the header decompressor 903. The header decompressor 903 refers to the reference information in the reference information manager 906 to decompress the compressed header, and outputs the resultant packet to the error detector 902. The error detector 902 detects any error in the header-decompressed packet, and outputs, to the packet output unit 901, any packet whose header has been correctly decompressed. The decom-

pression error detection result storage 905 counts the number of errors or successes detected in the packet by the error detector 902 for storage. The update request unit 907 is supplied with the number counted by the decompression error detection result storage 905, determining whether an update request is required or not and transmitting the update request as required to the transmitting side. The reference information manager 906 manages the reference information for header decompression.

[0056] Described specifically below is the operation of the above structured header decompression apparatus according to the present embodiment. Note that the operation of the header compression apparatus 607, that is, the transmitting side, is similar to that according to the first embodiment. Also assume that a packet supplied to the apparatuses contains data such as video and audio with an RTP/UDP/IP header.

[0057] In a header decompression apparatus 908 of FIG. 4, a packet receiver 904 receives a header-compressed packet transmitted from the packet transmitter 604 of FIG. 1, and outputs the packet to the header decompressor 903.

[0058] The header decompressor 903 refers to the reference information stored in the reference information manager 906 to decompress the compressed-header of the packet. The header-decompressed packet is outputted to the error detector 902. If the header is provided with update information of the reference information, the header decompressor 903 outputs the update information to the reference information manager 906.

[0059] The error detector 902 checks the presence or absence of an error in the header-decompressed packet by using the CRC. Also, the error detector 902 notifies the detection result to the decompression error detection result storage 905. If detecting any error, the error detector 902 discards the packet. If not detecting, the error detector 902 outputs, to the packet output unit 901, the packet with the CRC removed therefrom. The packet output unit 901 externally outputs the error-undetected RTP/UDP/IP packet.

[0060] The reference information manager 906 stores and manages the reference information required for header decompression. If the compressed header includes the update information, the reference information manager 906 updates the stored reference information with the update information coming from the header decompressor 903.

[0061] The decompression error detection result storage 905 stores the detection results of the error detector 902, that is, the number of errors R for preceding W packets. Instead of W or R , the number of decompression successes may be stored.

[0062] The update request unit 907 determines whether to request update by comparing the values of W and R outputted from the decompression error detection result storage 905 with predetermined values. For

example, if, among received 50 packets, 45 packets have errors, that is, $W = 50$ and $R \geq 45$, the update request unit 907 determines that the reference information has errors equal to or more than a predetermined value, and requests the transmitting side to update the reference information. Note that the above values are example only, and not restrictive.

[0063] As described above, the header decompression apparatus 908 according to the present embodiment requests updating of the reference information when the number of decompression errors becomes relatively or substantially large in the packets previously received by the receiving side. Thus, the number of packets discarded due to error in the reference information can be reduced, and efficient packet transmission can be achieved. Such capability of the header decompression apparatus 908 can be achieved in a general-purpose computer system. In this case, this capability is realized by a program executed in the computer system. The program is typically stored in a recoding medium such as CD-ROM, or transmitted through a communication medium.

[0064] While the invention has been described in detail, the foregoing description is in all aspects illustrative and not restrictive. It is understood that numerous other modifications and variations can be devised without departing from the scope of the invention.

Claims

1. A header decompression apparatus (709, 908) for decompressing a compressed header of a packet for transmission by referring to reference information being the same as reference information referred to for header compression by a transmitting side, said apparatus (709, 908) comprising:

a packet receiver (704, 904) for receiving the packet from said transmitting side;
 a reference information manager (707, 906) for storing and managing said reference information;
 a header decompressor (703, 903), provided with the received packet, for carrying out header decompression by referring to the reference information stored in said reference information manager (707, 906);
 an error detector (702, 902) for detecting an error in the packet including the decompressed header; and
 an update request unit (708, 907) for transmitting, to said transmitting side, an update request for requesting update information for updating said reference information, wherein when said update information is transmitted from said transmitting side, said reference information manager (707, 906) updates the ref-

erence information stored in said reference information manager (707, 906) based on the transmitted update information,

characterized in that

it is further comprised a counter/storage (705, 706, 905) for counting the number of errors detected by said error detector (702, 902), said counter/storage (705, 706) including

a successive error counter (705) for counting and storing the number of times X a packet having an error detected by said error detector (702) successively appeared; and

a successive decompression success counter (706) for counting and storing the number of times Y a packet not having an error detected by said error detector successively after counting by said successive error counter (705) is stopped; and
in that

said update request unit (708, 907) transmits said update request when determining, based on values of X and Y counted by said counter/storage (705, 706), that the reference information stored in said reference information manager (707, 906) should be updated.

2. The header decompression apparatus according to claim 1, wherein when $X \geq Y$, said update request unit (708) determines that the reference information stored in said reference information manager (707) should be updated.
3. A header decompression method for decompressing a compressed header of a packet for transmission by referring to reference information that is the same as reference information referred to for header compression by a transmitting side, said method comprising:

a packet receiving step of receiving the packet from said transmitting side;
 a header decompressing step, provided with the received packet, of carrying out header decompression by referring to the reference information stored in said reference information stored;
 an error detecting step of detecting error in the packet including the decompressed header; and
 an update requesting step of transmitting, to said transmitting side, an update request for requesting update information for updating said reference information;

characterized in

further comprising a counting/storing step of counting the number of errors detected in said error detecting step, said counting/storing step including

a successive error counting step of counting and storing the number of times X a packet having an error detected by said error detecting step successively appeared; and

a successive decompression success counting step of counting and storing the number of times Y a packet not having an error detected by said error detecting step successively appeared after counting by said successive error counting step is stopped; and **in that**

in said update requesting step said update request is transmitted when determining, based on values of X and Y counted in said counter/storage step, that the stored reference information should be updated.

4. The header decompression method according to claim 3, wherein in said update requesting step, when $X \geq Y$, it is determined that the stored reference information should be updated.

5. A computer-readable recording medium with a program recorded therein, the program being executed in a computer system for carrying out header decompression of decompressing a compressed header of a packet for transmission by referring to reference information that is the same as reference information referred to for header compression by a transmitting side, said program comprising:

a packet receiving step of receiving the packet from said transmitting side;

a header decompressing step, provided with the received packet, of carrying out header decompression by referring to the reference information stored in said reference information stored;

an error detecting step of detecting an error in the packet including the decompressed header; and

an update requesting step of transmitting, to said transmitting side, an update request for requesting update information for updating said reference information;

characterized in

further comprising a counting/storing step of counting the number of errors detected in said error detecting step, said counting/storing step including

a successive error counting step of counting and storing the number of times X a packet having an error detected by said error detecting step successively appeared; and

a successive decompression success counting step of counting and storing the number of times Y a packet not having an error detected by said error detecting step successively appeared after counting by said successive error counting step is

stopped; and **in that** in said

update requesting step said update request is transmitted, when determining, based on values of X and Y counted in said counter/storage step, that the stored reference information should be updated.

6. A computer program product for carrying out header decompression of decompressing a compressed header of a packet for transmission by referring to reference information the same as reference information referred to for header compression by a transmitting side, said program comprising code means adapted to perform, when said program is run on a data-processing system, the method steps of:

a packet receiving step of receiving the packet from said transmitting side;

a header decompressing step, provided with the received packet, of carrying out header decompression by referring to the reference information stored in said reference information stored;

an error detecting step of detecting an error in the packet including the decompressed header; and

an update requesting step of transmitting, to said transmitting side, an update request for requesting update information for updating said reference information;

characterized in

further comprising a counting/storing step of counting the number of errors detected in said error detecting step, said counting/storing step including

a successive error counting step of counting and storing the number of times X a packet having an error detected by said error detecting step successively appeared; and

a successive decompression success counting step of counting and storing the number of times Y a packet not having an error detected by said error detecting step successively appeared after counting by said successive error counting step is stopped; and **in that**

in said update requesting step said update request is transmitted, when determining, based on values of X and Y counted in said counter/storage step, that the stored reference information should be updated.

Patentansprüche

1. Header-Dekomprimierungsvorrichtung (709, 908) zum Dekomprimieren eines komprimierten Headers eines Pakets zur Übertragung durch Bezug-

nehmen auf Referenzinformation, welche dieselbe ist wie Referenzinformation, auf welche sich bezogen wird zur Header-Komprimierung von einer übertragenden Seite, wobei die Vorrichtung (709, 908) umfasst:

einen Paketempfänger (704, 904) zum Empfangen des Pakets von der übertragenden Seite;

einen Referenzinformationsverwalter (707, 906) zum Speichern und Verwalten der Referenzinformation;

einen Header-Dekomprimierer (703, 903), versehen mit dem empfangenen Paket, zum Ausführen von Header-Dekomprimierung durch Bezugnehmen auf die Referenzinformation, die in dem Referenzinformationsverwalter (707, 906) gespeichert ist;

einen Fehlererkenner (702, 902) zum Erkennen eines Fehlers in dem Paket, welches den dekomprimierten Header enthält; und

eine Aktualisierungsanforderungseinheit (708, 907) zum Übertragen, an die Übertragungsseite, einer Aktualisierungsanforderung zum Anfordern von Aktualisierungsinformation zum Aktualisieren der Referenzinformation, worin wenn die Aktualisierungsinformation von der Übertragungsseite übertragen wird, der Referenzinformationsverwalter (707, 906) die in dem Referenzinformationsverwalter (707, 906) gespeicherte Referenzinformation basierend auf der übertragenen Aktualisierungsinformation aktualisiert,

dadurch gekennzeichnet, dass

weiterhin umfasst ist ein Zähler/Speicher (705, 706, 905) zum Zählen der Anzahl an Fehlern, die durch den Fehlererkenner (702, 902) erkannt wurden, wobei der Zähler/Speicher (705, 706) enthält

einen Zähler (705) für aufeinander folgende Fehler zum Zählen und Speichern der Anzahl an Malen X, wie oft ein Paket, bei dem durch den Fehlererkenner (702) ein Fehler erkannt wird, aufeinander folgend erschienen ist; und

einen Zähler für aufeinander folgenden Dekomprimierungserfolg (706) zum Zählen und Speichern der Anzahl an Malen Y, wie oft aufeinander folgend ein Paket, bei dem durch den Fehlererkenner kein Fehler erkannt wird, nachdem das Zählen durch den Zähler für aufeinander folgende Fehler (705) angehalten ist; und dadurch dass die Aktualisierungsanforderungseinheit (708, 907) die Aktualisierungsanforderung überträgt, wenn sie bestimmt, basierend auf Werten von X und Y, die durch den Zähler/Speicher (705, 706) gezählt wurden, dass die in dem Referenzinformationsverwalter (707, 906) gespeicherte Referenzinformation aktualisiert werden soll.

2. Header-Dekomprimierungsvorrichtung nach Anspruch 1, worin wenn $X \geq Y$, die Aktualisierungsanforderungseinheit (708) bestimmt, dass die Referenzinformation, die in dem Referenzinformationsverwalter (707) gespeichert ist, aktualisiert werden soll.

3. Header-Dekomprimierungsverfahren zum Dekomprimieren eines komprimierten Headers eines Pakets zur Übertragung durch Bezugnehmen auf Referenzinformation, welche dieselbe ist wie Referenzinformation, auf die sich zur Header-Komprimierung durch eine Übertragungsseite bezogen wird, wobei das Verfahren umfasst:

einen Paketempfangsschritt des Empfangens des Pakets von der Übertragungsseite;

einen Header-Dekomprimierungsschritt, versehen mit dem empfangenen Paket, des Ausführens von Header-Dekomprimierung durch Bezugnehmen auf die Referenzinformation in der gespeicherten Referenzinformation;

einen Fehlererkennungsschritt des Erkennens eines Fehlers in dem Paket, welches den dekomprimierten Header enthält; und

einen Aktualisierungsanforderungsschritt des Übertragens, an die Übertragungsseite, einer Aktualisierungsanforderung zum Anfordern von Aktualisierungsinformation zum Aktualisieren der Referenzinformation;

dadurch gekennzeichnet, dass

weiter umfasst ist ein Zähl-/Speicherschritt des Zählens der Anzahl an Fehlern, die in dem Fehlererkennungsschritt erkannt wurden, wobei der Zähl-/Speicherschritt enthält

einen Schritt des Zählens aufeinander folgender Fehler, des Zählens und Speicherns der Anzahl an Malen X, die ein Paket, das durch den Fehlererkennungsschritt einen Fehler erkannt hat, aufeinander folgend erschienen ist; und

einen Schritt des Zählens aufeinander folgender Dekomprimierungserfolge, des Zählens und Speicherns der Anzahl an Malen Y, die ein Paket, das keinen Fehler durch den Fehlererkennungsschritt erkannt hat, aufeinander folgend erschienen ist, nachdem das Zählen durch den Schritt des Zählens aufeinander folgender Fehler angehalten wurde; und dadurch, dass

in dem Aktualisierungsanforderungsschritt die Aktualisierungsanforderung übertragen wird, wenn bestimmt wird, basierend auf Werten von X und Y, die in dem Zähl-/Speicherschritt gezählt wurden, dass die gespeicherte Referenzinformation aktualisiert werden soll.

4. Header-Dekomprimierungsverfahren nach Anspruch 3, worin in dem Aktualisierungsanforde-

rungsschritt, wenn $X \geq Y$, bestimmt wird, dass die gespeicherte Referenzinformation aktualisiert werden soll.

5. Computer-lesbares Aufzeichnungsmedium, mit einem darin aufgezeichneten Programm, wobei das Programm in einem Computersystem ausgeführt wird zum Durchführen von Header-Dekomprimierung des Dekomprimierens eines komprimierten Headers eines Pakets zur Übertragung durch Bezugnehmen auf Referenzinformation, welche dieselbe ist wie Referenzinformation, auf welche sich zur Header-Komprimierung durch eine Übertragungsseite bezogen wird, wobei das Programm umfasst:

einen Paketempfangsschritt des Empfangens des Pakets von der Übertragungsseite;
einen Header-Dekomprimierungsschritt, versehen mit dem empfangenen Paket, des Ausführens von Header-Dekomprimierung durch Bezugnehmen auf die Referenzinformation in der gespeicherten Referenzinformation;
einen Fehlererkennungsschritt des Erkennens eines Fehlers in dem Paket, welches den dekomprimierten Header enthält; und
einen Aktualisierungsanforderungsschritt des Übertragens, an die Übertragungsseite, einer Aktualisierungsanforderung zum Anfordern von Aktualisierungsinformation zum Aktualisieren der Referenzinformation;

dadurch gekennzeichnet, dass

weiter umfasst ist ein Zähl-/Speicherschritt des Zählens der Anzahl an Fehlern, die in dem Fehlererkennungsschritt erkannt wurden, wobei der Zähl-/Speicherschritt enthält

einen Schritt des Zählens aufeinander folgender Fehler des Zählens und Speicherns der Anzahl an Malen X, die ein Paket, das durch den Fehlererkennungsschritt einen Fehler erkannt hat, aufeinander folgend erschienen ist; und

einen Schritt des Zählens aufeinander folgender Dekomprimierungserfolge des Zählens und Speicherns der Anzahl an Malen Y, die ein Paket, das keinen Fehler durch den Fehlererkennungsschritt erkannt hat, aufeinander folgend erschienen ist, nachdem das Zählen durch den Schritt des Zählens aufeinander folgender Fehler angehalten wurde; und dadurch, dass

in dem Aktualisierungsanforderungsschritt die Aktualisierungsanforderung übertragen wird, wenn bestimmt wird, basierend auf Werten von X und Y, die in dem Zähl-/Speicherschritt gezählt wurden, dass die gespeicherte Referenzinformation aktualisiert werden soll.

6. Computerprogrammprodukt zum Durchführen von

Header-Dekomprimierung des Dekomprimierens eines komprimierten Headers eines Pakets zur Übertragung durch Bezugnehmen auf Referenzinformation, die gleich ist wie Referenzinformation, auf die sich zur Header-Komprimierung durch eine Übertragungsseite bezogen wird, wobei das Programm Codemittel umfasst, die angepasst sind, wenn das Programm auf einem Datenverarbeitungssystem ausgeführt wird, die Verfahrensschritte durchzuführen:

einen Paketempfangsschritt des Empfangens des Pakets von der Übertragungsseite;
einen Header-Dekomprimierungsschritt, versehen mit dem empfangenen Paket, des Ausführens von Header-Dekomprimierung durch Bezugnehmen auf die Referenzinformation in der gespeicherten Referenzinformation;
einen Fehlererkennungsschritt des Erkennens eines Fehlers in dem Paket, welches den dekomprimierten Header enthält; und
einen Aktualisierungsanforderungsschritt des Übertragens, an die Übertragungsseite, einer Aktualisierungsanforderung zum Anfordern von Aktualisierungsinformation zum Aktualisieren der Referenzinformation;

dadurch gekennzeichnet, dass

weiter umfasst ist ein Zähl-/Speicherschritt des Zählens der Anzahl an Fehlern, die in dem Fehlererkennungsschritt erkannt wurden, wobei der Zähl-/Speicherschritt enthält

einen Schritt des Zählens aufeinander folgender Fehler des Zählens und Speicherns der Anzahl an Malen X, die ein Paket, das durch den Fehlererkennungsschritt einen Fehler erkannt hat, aufeinander folgend erschienen ist; und

einen Schritt des Zählens aufeinander folgender Dekomprimierungserfolge des Zählens und Speicherns der Anzahl an Malen Y, die ein Paket, das keinen Fehler durch den Fehlererkennungsschritt erkannt hat, aufeinander folgend erschienen ist, nachdem das Zählen durch den Schritt des Zählens aufeinander folgender Fehler angehalten wurde; und dadurch, dass

in dem Aktualisierungsanforderungsschritt die Aktualisierungsanforderung übertragen wird, wenn bestimmt wird, basierend auf Werten von X und Y, die in dem Zähl-/Speicherschritt gezählt wurden, dass die gespeicherte Referenzinformation aktualisiert werden soll.

Revendications

1. Appareil de décompression d'en-tête (709, 908) pour décompresser un en-tête compressé d'un paquet pour transmission en se référant à des infor-

mations de référence étant les mêmes que les informations de référence auxquelles il s'est référé pour la compression d'en-tête par un côté émission, ledit appareil (709, 908) comprenant :

un récepteur de paquet (704, 904) pour recevoir le paquet dudit côté émission ;

un gestionnaire d'informations de référence (707, 906) pour mémoriser et gérer lesdites informations de référence ;

un décompresseur d'en-tête (703, 903), pourvu du paquet reçu, pour effectuer la décompression d'en-tête en se référant aux informations de référence mémorisées dans ledit gestionnaire d'informations de référence (707, 906) ;

un détecteur d'erreur (702, 902) pour détecter une erreur dans le paquet incluant l'en-tête décompressée ; et

une unité de demande de mise à jour (708, 907) pour transmettre, vers ledit côté émission, une demande de mise à jour pour demander des informations de mise à jour pour mettre à jour lesdites informations de référence, dans lequel

lorsque lesdites informations de mise à jour sont transmises depuis ledit côté émission, ledit gestionnaire d'informations de référence (707, 906) met à jour les informations de référence mémorisées dans ledit gestionnaire d'informations de référence (707, 906) sur la base des informations de mise à jour transmises,

caractérisé en ce que

il comprend, en outre, un compteur/mémoire (705, 706, 905) pour compter le nombre d'erreurs détectées par ledit détecteur d'erreur (702, 902), ledit compteur/mémoire (705, 706) incluant

un compteur d'erreurs successives (705) pour compter et mémoriser le nombre de fois X où un paquet ayant une erreur détectée par ledit détecteur (702) apparaît successivement ; et

un compteur de décompression successive satisfaisante (706) pour compter et mémoriser le nombre de fois Y où un paquet n'ayant pas d'erreur détectée par ledit détecteur d'erreur apparaît successivement après que le comptage par ledit compteur d'erreurs successives (705) soit arrêté ; et **en ce que**

ladite unité de demande de mise à jour (708, 907) transmet ladite demande de mise à jour lors de la détermination, sur la base des valeurs de X et Y comptées par ledit compteur/mémoire (705, 706), que les informations de référence mémorisées dans ledit gestionnaire d'informations de référence (707, 906) devraient être mises jour.

2. Appareil de décompression d'en-tête selon la revendication 1, dans lequel lorsque $X \geq Y$, ladite unité de demande de mise à jour (708) détermine que les informations de référence mémorisées dans ledit gestionnaire d'informations de référence (707) devraient être mises à jour.

3. Procédé de décompression d'en-tête pour décompresser un en-tête compressé d'un paquet pour transmission en se référant aux informations de référence qui sont les mêmes que les informations de référence auxquelles il s'est référé pour la compression d'en-tête par un côté émission, ledit procédé comprenant :

une étape de réception de paquet consistant à recevoir le paquet dudit côté émission ;

une étape de décompression d'en-tête, pourvu du paquet reçu, consistant à effectuer la décompression d'en-tête en se référant aux informations de référence mémorisées dans ledit gestionnaire d'informations de référence ;

une étape de détection d'erreur consistant à détecter l'erreur dans le paquet incluant l'en-tête décompressée ; et

une étape de demande de mise à jour consistant à transmettre, audit côté émission, une demande de mise à jour pour demander des informations de mise à jour pour mettre à jour lesdites informations de référence ;

caractérisé en ce que

il comprend, en outre, une étape de comptage/mémorisation consistant à compter le nombre d'erreurs détectées dans ladite étape de détection d'erreur, ladite étape de comptage/mémorisation incluant

une étape de comptage d'erreurs successives consistant à compter et mémoriser le nombre de fois X où un paquet ayant une erreur détectée par ladite étape de détection d'erreur apparaît successivement ; et

une étape de comptage de décompression successive satisfaisante consistant à compter et à mémoriser le nombre de fois Y où un paquet n'ayant pas d'erreur détectée par l'étape de détection d'erreur apparaît successivement après que le comptage par ladite étape de comptage d'erreur successive soit arrêté ; et **en ce que**

dans ladite étape de demande de mise à jour, ladite demande de mise à jour est transmise lors de la détermination, sur la base des valeurs de X et Y comptées dans ladite étape de comptage/mémorisation, que les informations de référence mémorisées devraient être mises jour.

4. Procédé de décompression d'en-tête selon la revendication 3, dans lequel dans ladite étape de demande de mise à jour, lorsque $X \geq Y$, il est déterminé que les informations de référence mémorisées devraient être mises à jour. 5
5. Support d'enregistrement lisible par ordinateur avec un programme enregistré dans celui-ci, le programme étant exécuté dans un système d'ordinateur pour effectuer la décompression d'en-tête de la décompression d'un en-tête compressé d'un paquet pour transmission en se référant aux informations de référence qui sont les mêmes que les informations de référence auxquelles il s'est référé pour la compression d'en-tête par un côté émission, ledit programme comprenant : 10
- une étape de réception de paquet consistant à recevoir le paquet dudit côté émission ; 15
- une étape de décompression d'en-tête, pourvu du paquet reçu, consistant à effectuer la décompression d'en-tête en se référant aux informations de référence mémorisées dans ledit gestionnaire d'informations de référence ; 20
- une étape de détection d'erreur consistant à détecter une erreur dans le paquet incluant l'en-tête décompressée ; et 25
- une étape de demande de mise à jour consistant à transmettre, vers ledit côté émission, une demande de mise à jour pour demander des informations de mise à jour pour mettre à jour lesdites informations de référence ; 30
- caractérisé en ce que**
- il comprend, en outre, une étape de comptage/mémorisation consistant à compter le nombre d'erreurs détectées dans ladite étape de détection d'erreur, ladite étape de comptage/mémorisation incluant 35
- une étape de comptage d'erreurs successives consistant à compter et mémoriser le nombre de fois X où un paquet ayant une erreur détectée par ladite étape de détection d'erreur apparaît successivement ; et 40
- une étape de comptage de décompression successive satisfaisante consistant à compter et à mémoriser le nombre de fois Y où un paquet n'ayant pas d'erreur détectée par ladite étape de détection d'erreur apparaît successivement après que le comptage par ladite étape de comptage d'erreur successive soit arrêté ; et **en ce que** 45
- dans ladite étape de demande de mise à jour, ladite demande de mise à jour est transmise lors de la détermination, sur la base des valeurs de X et Y comptées dans ladite étape de comptage/mémorisation, que les informations de référence mémorisées devraient être mises à jour. 50

sation, que les informations de référence mémorisées devraient être mises à jour.

6. Produit de programme d'ordinateur pour effectuer la décompression d'en-tête consistant à décompresser un en-tête compressé d'un paquet pour transmission en se référant à des informations de référence qui sont les mêmes que les informations de référence auxquelles il s'est référé pour la compression d'en-tête par un côté émission, ledit programme comprenant un moyen de code conçu pour effectuer, lorsque ledit programme est exécuté sur un système de traitement de données, les étapes du procédé de : 5

une étape de réception de paquet consistant à recevoir le paquet dudit côté émission ;

une étape de décompression d'en-tête, pourvu du paquet reçu, consistant à effectuer la décompression d'en-tête en se référant aux informations de référence mémorisées dans ledit gestionnaire d'informations de référence ;

une étape de détection d'erreur consistant à détecter une erreur dans le paquet incluant l'en-tête décompressée ; et

une étape de demande de mise à jour consistant à transmettre, vers ledit côté émission, une demande de mise à jour pour demander les informations de mise à jour pour mettre à jour lesdites informations de référence ;

caractérisé en ce que

il comprend, en outre, une étape de comptage/mémorisation consistant à compter le nombre d'erreurs détectées dans ladite étape de détection d'erreur, ladite étape de comptage/mémorisation incluant

une étape de comptage d'erreurs successives consistant à compter et mémoriser le nombre de fois X où un paquet ayant une erreur détectée par ladite étape de détection d'erreur apparaît successivement ; et

une étape de comptage de décompression successive satisfaisante consistant à compter et à mémoriser le nombre de fois Y où un paquet n'ayant pas une erreur détectée par ladite étape de détection d'erreur apparaît successivement après que le comptage par ladite étape de comptage d'erreur successive soit arrêté ; et **en ce que**

dans ladite étape de demande de mise à jour, ladite demande de mise à jour est transmise lors de la détermination, sur la base des valeurs de X et Y comptées dans ladite étape de comptage/mémorisation, que les informations de référence mémorisées devraient être mises à jour.

FIG. 1

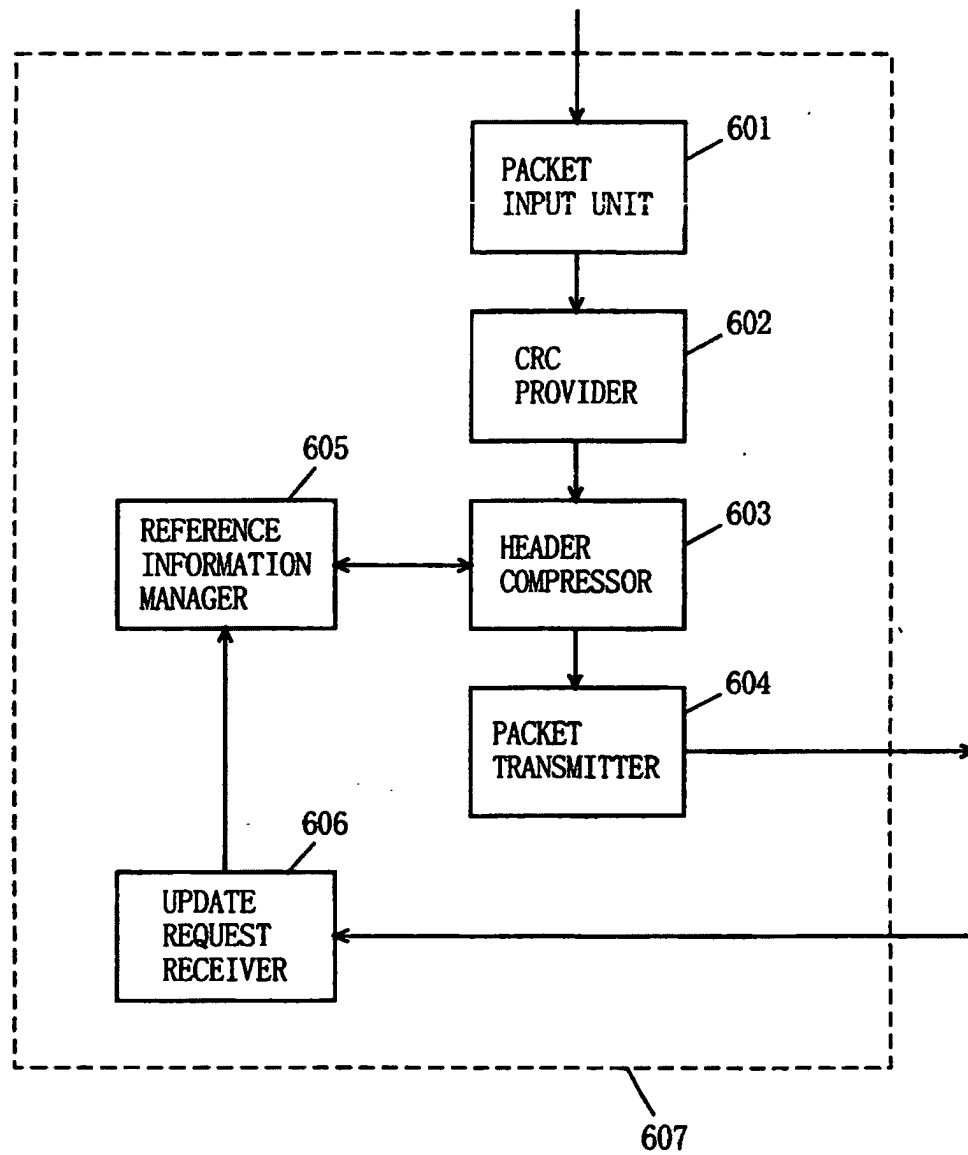


FIG. 2

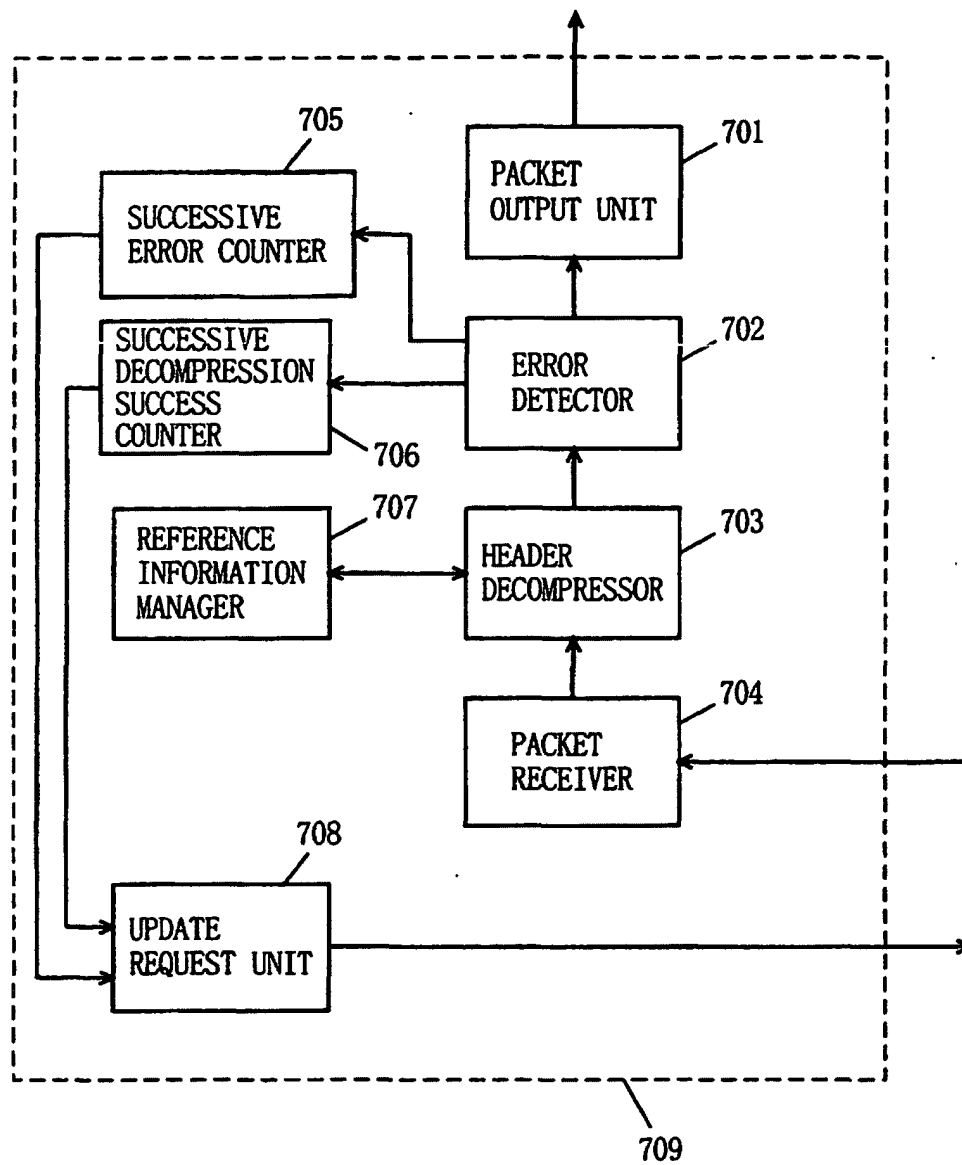


FIG. 3

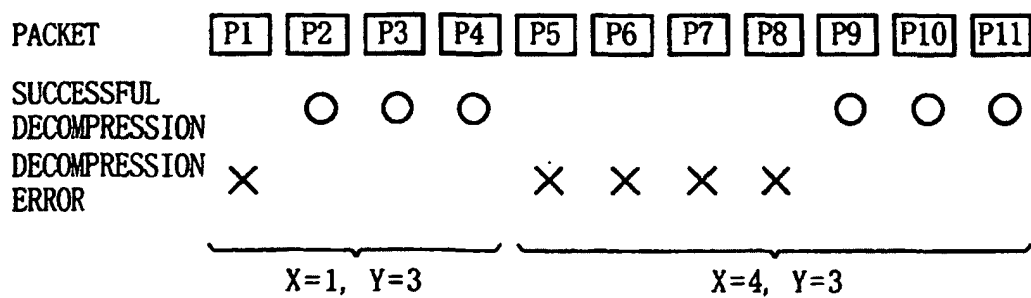


FIG. 4

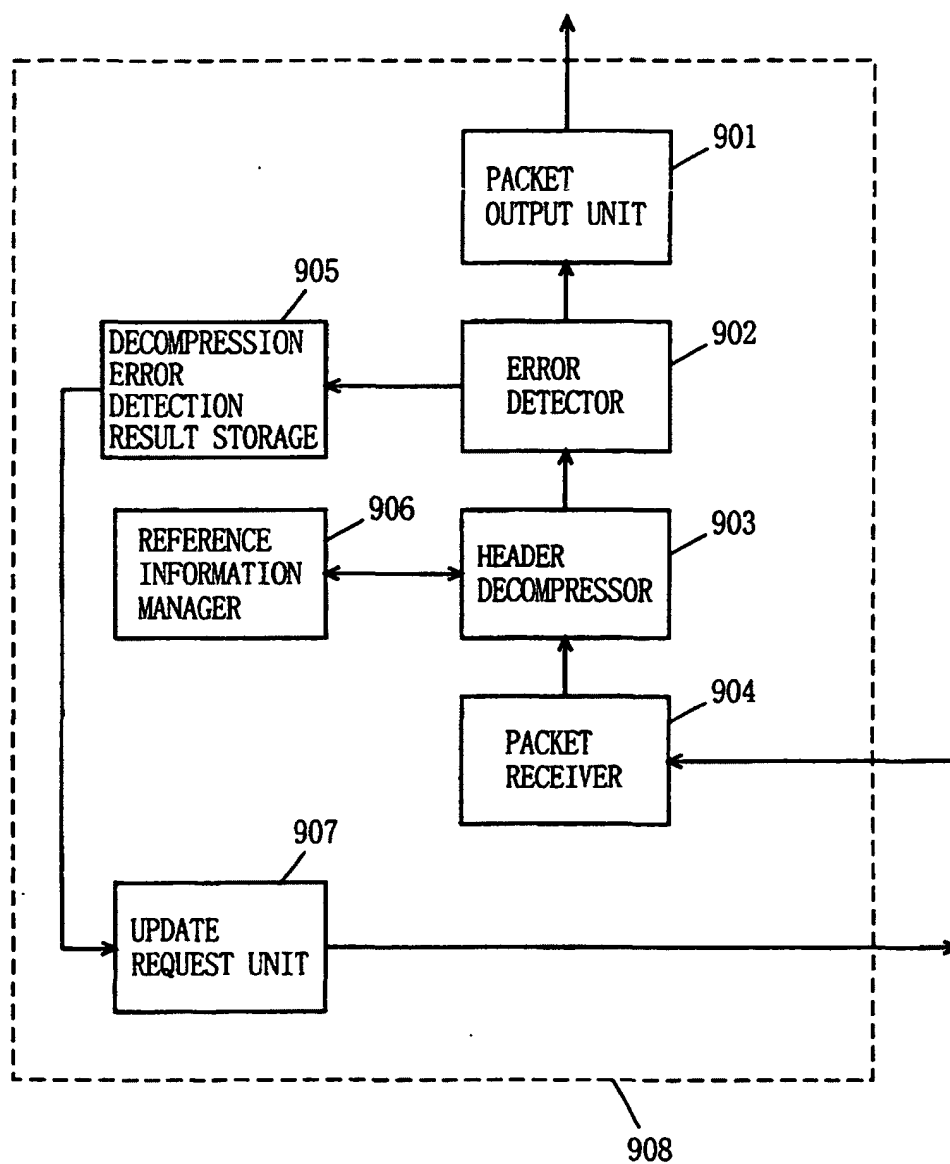


FIG. 5

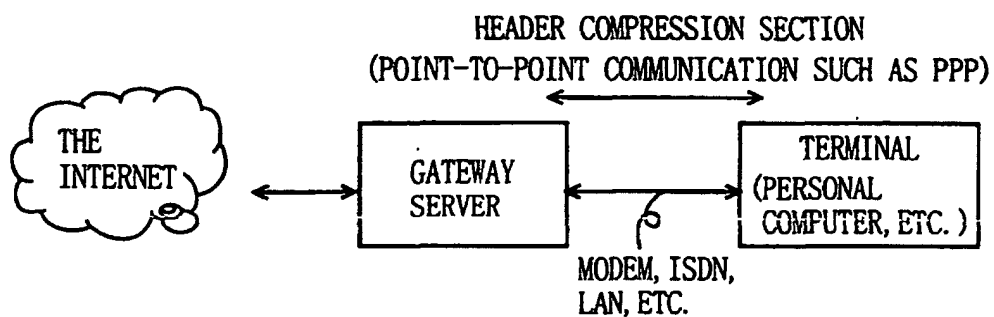


FIG. 6

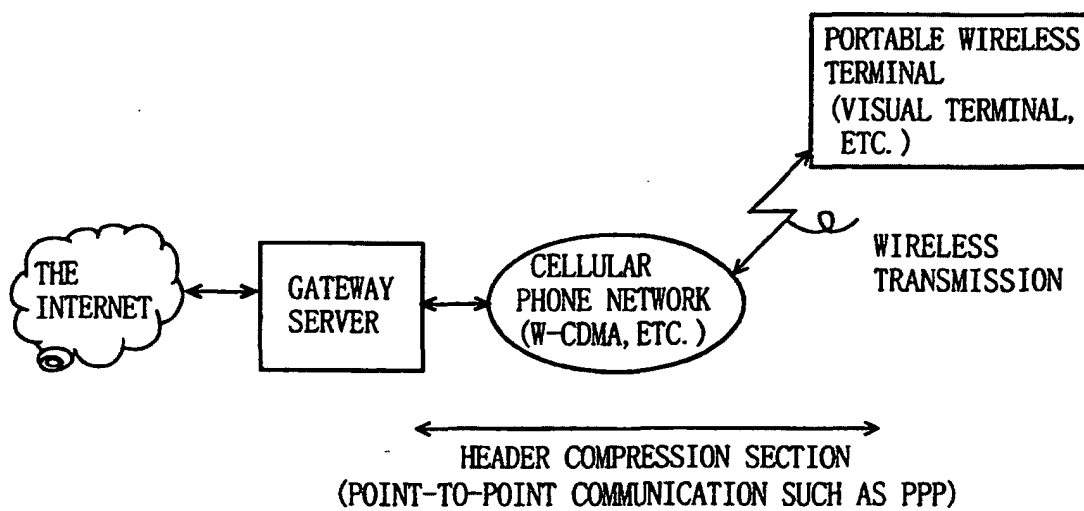


FIG. 7

TRANSMITTING SIDE
(HEADER COMPRESSION)

RECEIVING SIDE
(HEADER DECOMPRESSION)

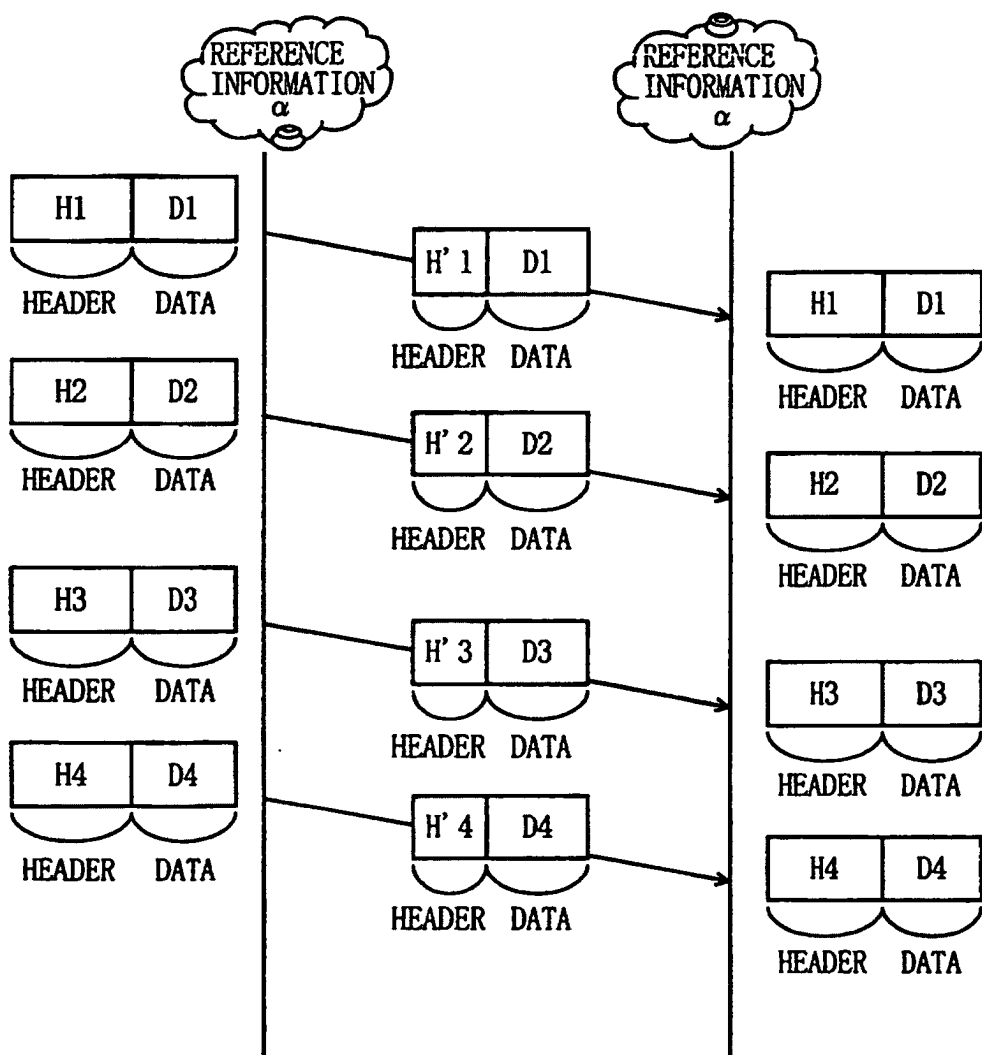


FIG. 8

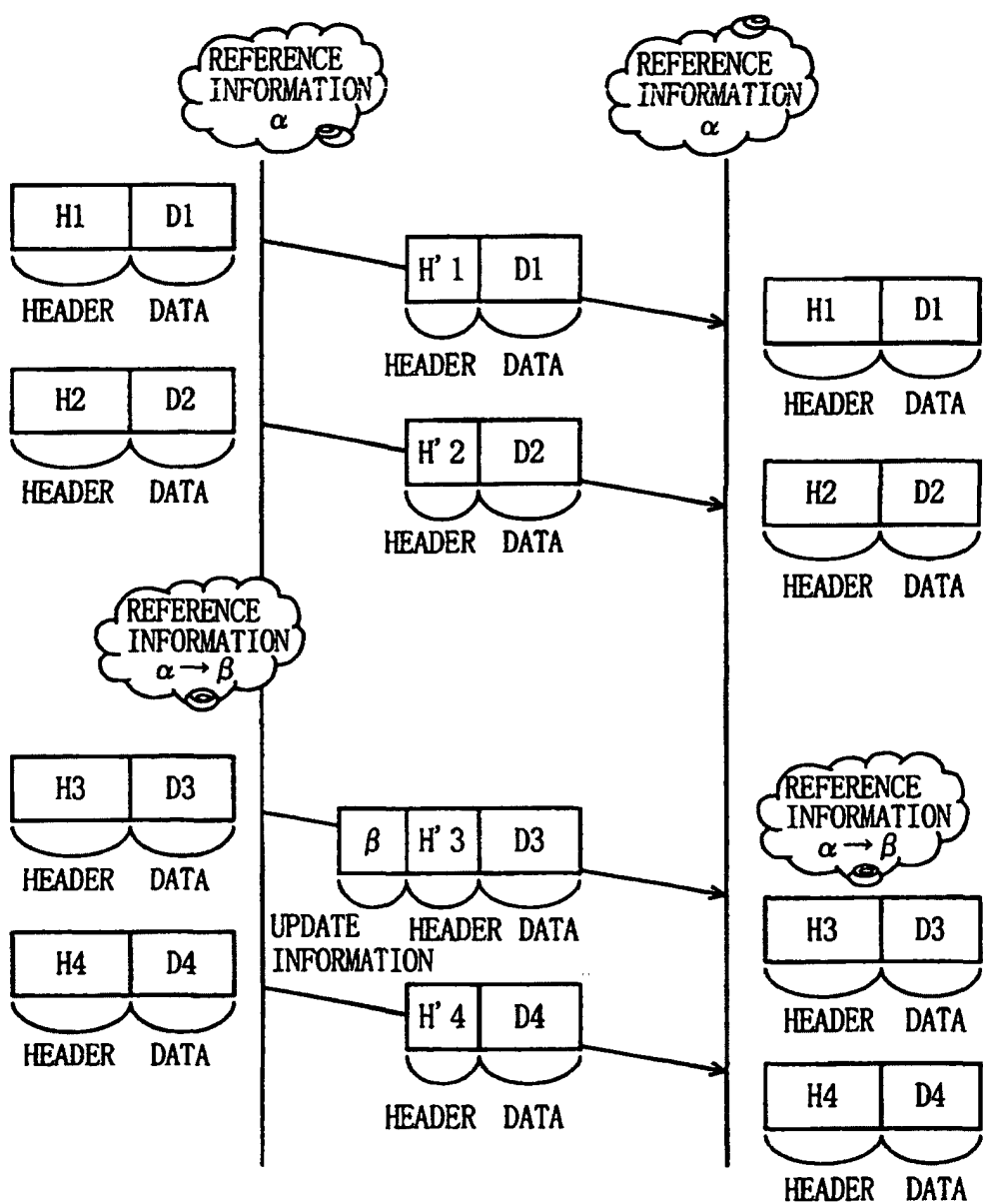
TRANSMITTING SIDE
(HEADER COMPRESSION)RECEIVING SIDE
(HEADER DECOMPRESSION)

FIG. 9

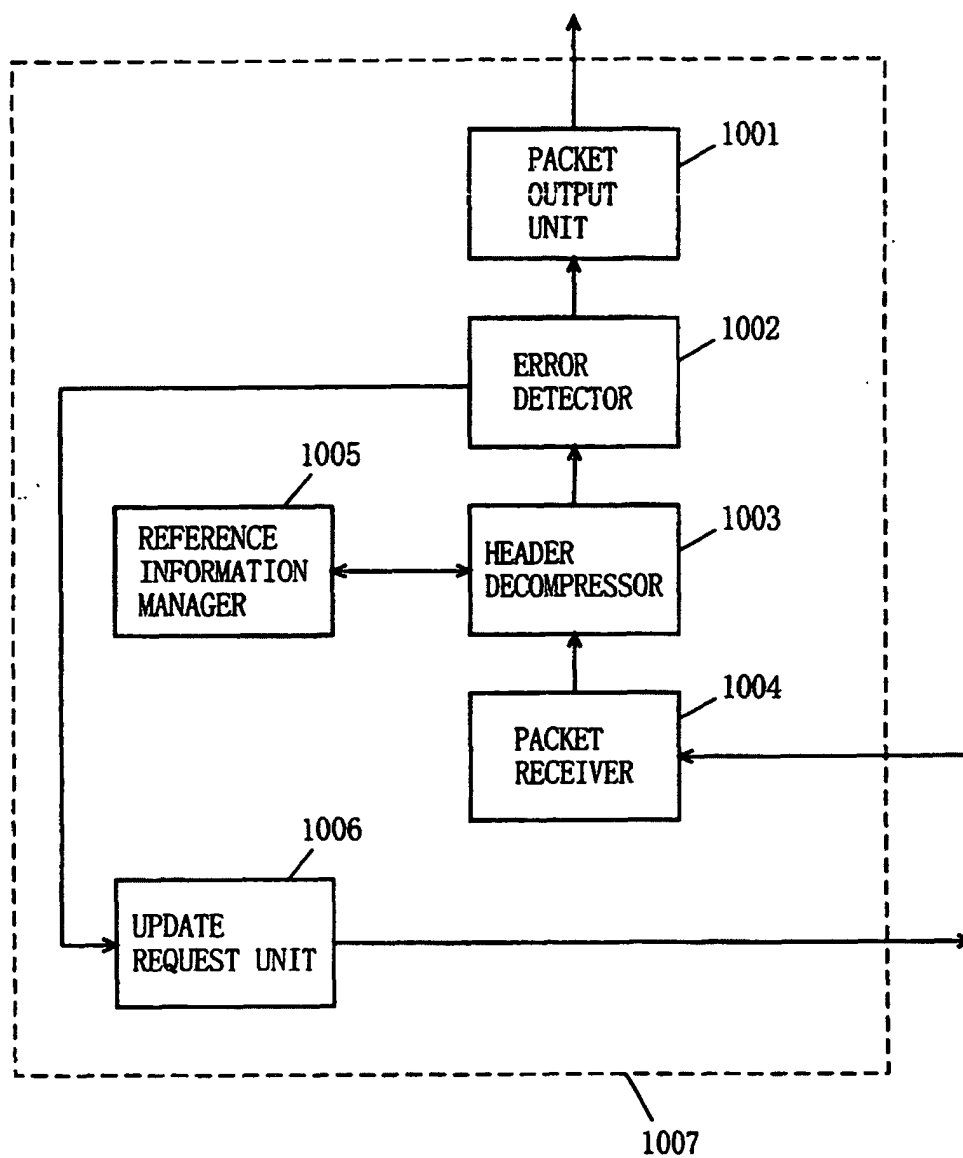


FIG. 10

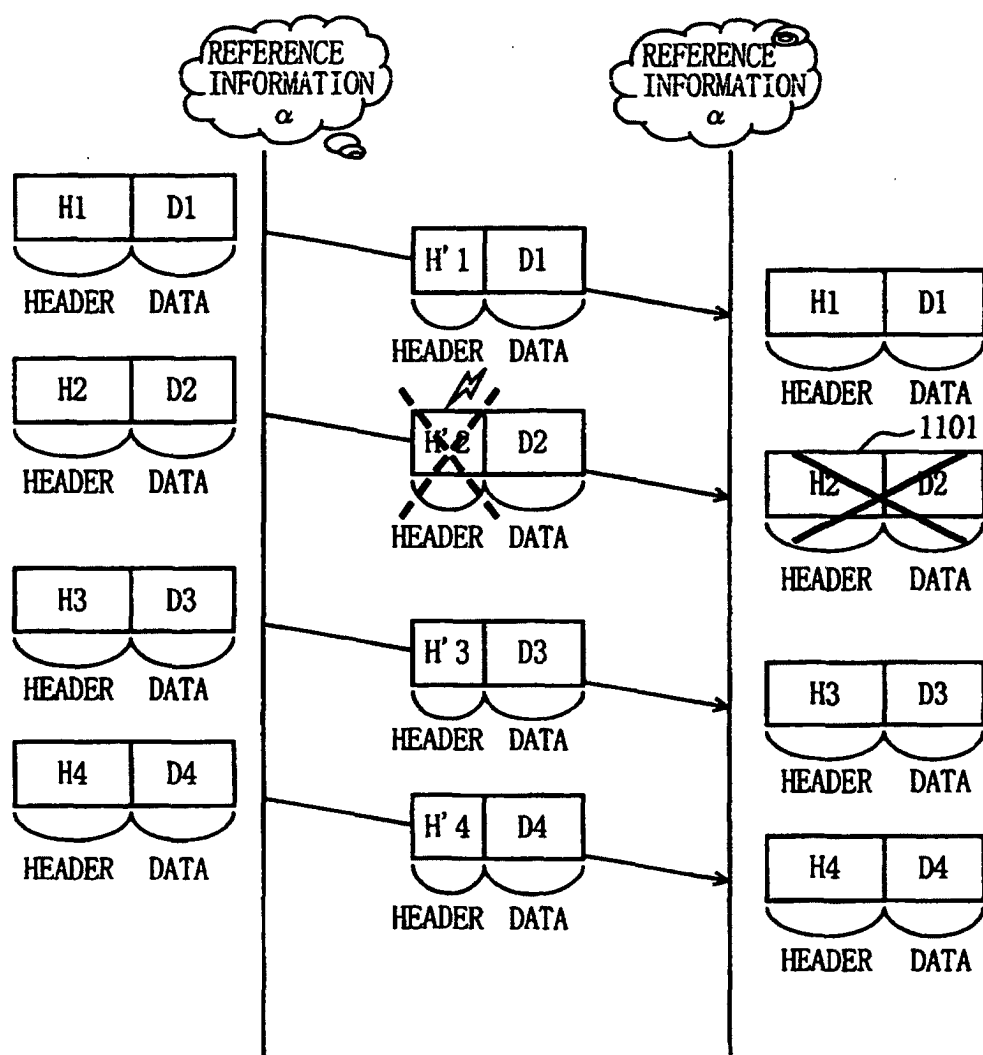
TRANSMITTING SIDE
(HEADER COMPRESSION)RECEIVING SIDE
(HEADER DECOMPRESSION)

FIG. 11

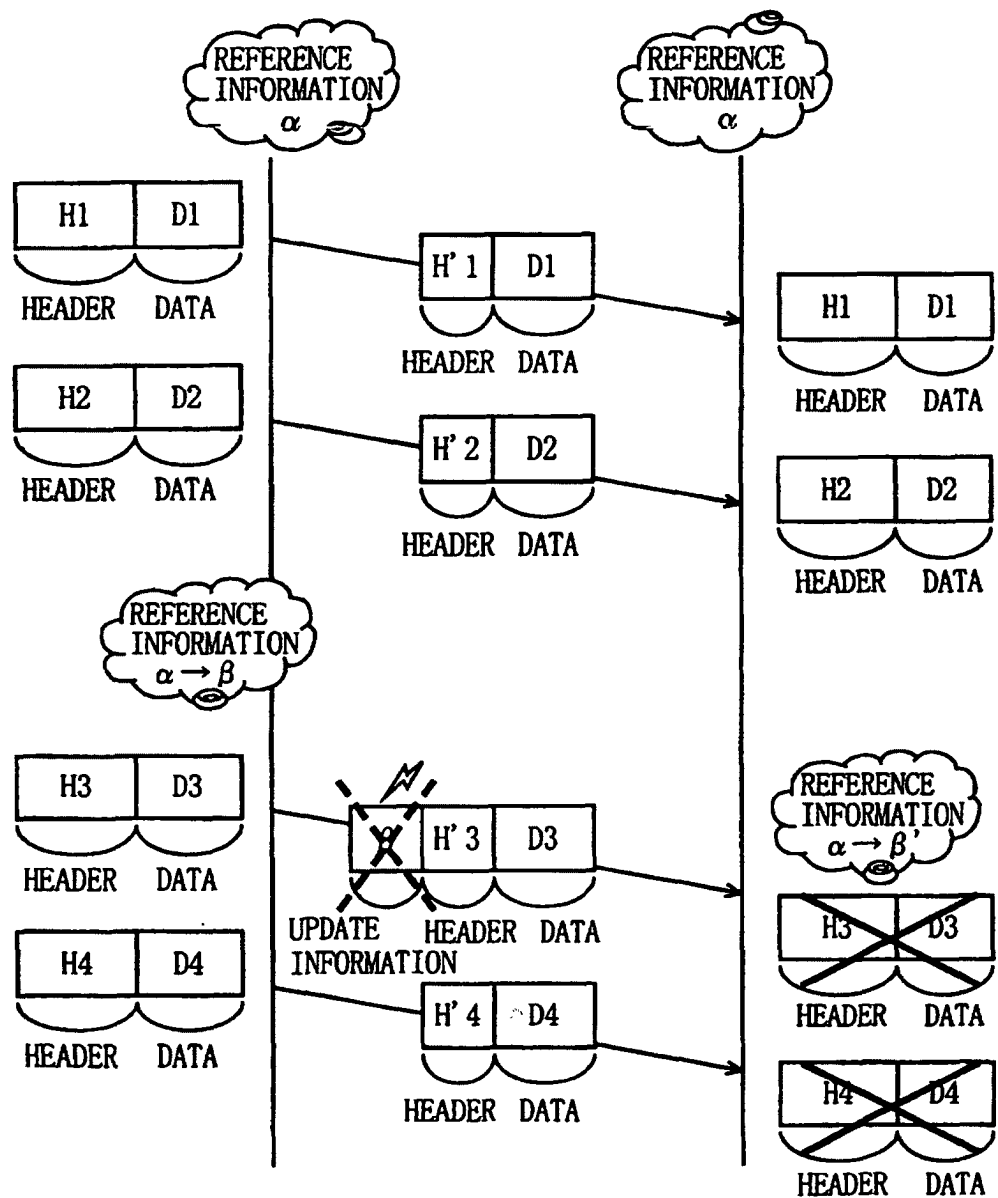
TRANSMITTING SIDE
(HEADER COMPRESSION)RECEIVING SIDE
(HEADER DECOMPRESSION)

FIG. 12

TRANSMITTING SIDE
(HEADER COMPRESSION)RECEIVING SIDE
(HEADER DECOMPRESSION)